ISSC 2009, UCD, June 10-11th

Automatic Hexaphonic Guitar Transcription Using Non-Negative Constraints

Paul D. O'Grady and Scott T. Rickard

Complex & Adaptive Systems Laboratory,
University College Dublin,
Ireland.

E-mail: paul.d.ogrady@ucd.ie

Abstract — **Automatic music transcription is a widely studied problem, Typically, recordings that are used for transcription are taken from standard instruments, in the case of electric stringed instruments—such as the electric guitar—the recordings are captured from a standard pick-up, which unwantedly mixes the signals from each string and complicates subsequent analysis. We propose an approach to electric guitar transcription where the signal generated by each string at the guitar pickup is captured and analysed separately; thus providing six separate signals as opposed to one mixed signal, which enables finger positions to be identified. Such an instrument is known as a hexaphonic guitar and is a popular instrument for spatial music performances. We build the equipment necessary to modify a standard electric guitar into a hexaphonic guitar, and present an application of Non-Negative Matrix Factorisation to the task of transcription—where a basis for each note on the fretboard is learned and fitted to a magnitude spectrogram of the hexaphonic recording, which then undergoes a non-linearity generating a piano roll representation of the music performance.**

Keywords — **Hexaphonic Guitar, Automatic Music Transcription, Non-Negative Matrix Factorisation.**

I Introduction

Music transcription is the act of notating a piece of music whereby the musician transcribes, by hand, the sequence of musical events that describe the piece. However, many musicians can not read or write musical notation, and employ a music transcriber to transfer the music to human readable form. Usually the music is transcribed by ear, where the skills required by the transcriber include *perfect pitch*, *i.e.* identify a musical note without assistance from an external reference.

Automatic music transcription endeavours to replace the human transcriber with a machine that perfectly transcribes the music into a machine or human readable form. Like human transcription, automatic music transcription requires pitch identification. Furthermore, tasks that are simple for a human, *e.g.* onset detection, instrument detection, vibrato, playing style *etc.*, are much more difficult for a machine. Consequently, automatic music transcription remains an open problem.

Many automatic music transcription approaches have been proposed over the last number of years including probabilistic methods for note onset detection [1] and methods related to Non-Negative Matrix Factorisation [2] and sparse coding [3]. Typically, automatic transcription algorithms are applied to recordings taken from standard instruments by standard recording equipment. In the case of the electric guitar, each string is sensed *separately* by a magnetic polepiece, the signals from each are summed together and connected to the output jack, from which the recording is captured. However, for automatic transcription, polyphonic music (as captured at the output jack) is much more difficult to transcribe than monophonic music (as captured for each string at the pickup). So it is fair to say that for the purposes of automatic guitar transcription, the string signals are unwantedly summed together at the pickup.

In this paper, we propose an alternative paradigm to electric guitar transcription, where the signal generated by each string at the guitar pickup is captured separately using a *hexaphonic guitar*; thus changing a polyphonic transcription problem into a monophonic one, which ameliorates subsequent analysis and allows finger position identification for tablature generation. We build the equipment necessary to modify a standard electric guitar into a hexaphonic guitar (as outlined in Section IV) and employ methods related to Non-Negative Matrix Factorisation in automatic transcription, where a fixed basis—which represents all notes on the fretboard—is learned from training data and is fitted to hexaphonic guitar recordings, resulting in a *piano roll* representation, which indicates the notes required to transcribe the music. Furthermore, for validation purposes, we create a MIDI file from the piano roll representation and listen to the results.

This paper is organised as follows: We present an overview of NMF in Section II and discuss its application to automatic hexaphonic guitar transcription in Section III. We overview the hexaphonic guitar modification and recording procedure in Section IV, and present a transcription example in Section V. Finally, we complete the paper with a discussion in Section VI.

II Non-negative Matrix Factorisation

Non-Negative Matrix Factorisation (NMF) [4] is a method for the decomposition of multivariate data, where a non-negative matrix, $\mathbf{V}$, is approximated as a product of two non-negative matrices, $\mathbf{V} \approx \mathbf{WH}$. NMF is a *parts-based* approach that makes no statistical assumption about the data. Instead, it assumes for the domain under consideration, *e.g.* magnitude spectrograms, that negative numbers are physically meaningless—which is the foundation for the assumption that the search for a decomposition should be confined to a non-negative space, *i.e.*, nonnegativity assumption. The lack of statistical assumptions makes it difficult to prove that NMF will give correct decompositions. However, it has been shown in practice to give correct results.

NMF, and its extensions, has been applied to a wide variety of problems including brain imaging [5], automatic ASCII art conversion [6] and speech separation [7, 8].

a) The NMF Algorithm

Given a non-negative matrix $\mathbf{V} \in \mathbb{R}^{\geq 0, M \times T}$, the goal is to approximate $\mathbf{V}$ as a product of two non-negative matrices $\mathbf{W} \in \mathbb{R}^{\geq 0, M \times R}$ and $\mathbf{H} \in \mathbb{R}^{\geq 0, R \times T}$, $\mathbf{V} \approx \mathbf{WH}, v_{ik} \approx \sum_{j=1}^{R} w_{ij} h_{jk}$. Typically, $R < M$, where $\mathbf{W}$ contains a low-rank basis and $\mathbf{H}$ contains associated activations.

Two NMF algorithms were introduced by Lee and Seung [4], each optimising a different cost function to measure reconstruction quality. The cost functions specified are the Squared Euclidean Distance (SED), $D_{\mathrm{SED}}(\mathbf{V}, \mathbf{W}, \mathbf{H}) = \frac{1}{2}\|\mathbf{V} - \mathbf{WH}\|^2$, and a generalised version of the Kullback-Leibler Divergence (KLD), $D_{\mathrm{KLD}}(\mathbf{V}\|\mathbf{W}, \mathbf{H}) = \sum_{ik}\Big(v_{ik} \log \frac{v_{ik}}{[\mathbf{WH}]_{ik}} - v_{ik} + [\mathbf{WH}]_{ik}\Big)$. Sometime after, Cichocki proposed the β-divergence as the NMF cost function [9], which is a parameterised divergence measure that encompasses both SED and KLD, and also includes the Itakura-Saito Divergence (ISD): $D_{\mathrm{ISD}}(\mathbf{V}\|\mathbf{W}, \mathbf{H}) = \sum_{ik}\Big(\frac{v_{ik}}{[\mathbf{WH}]_{ik}} - \log \frac{v_{ik}}{[\mathbf{WH}]_{ik}} - 1\Big)$. The NMF cost function utilising β-divergence is

$$D_{\mathrm{BD}}(\mathbf{V}\|\mathbf{W}, \mathbf{H}, \beta) = \sum_{ik}\Big(v_{ik} \frac{v_{ik}^{\beta-1} - [\mathbf{WH}]_{ik}^{\beta-1}}{\beta(\beta-1)} + [\mathbf{WH}]_{ik}^{\beta-1} \frac{[\mathbf{WH}]_{ik} - v_{ik}}{\beta}\Big), \tag{1}$$

for $\beta = 2$, SED is obtained; for $\beta \to 1$, the divergence tends to KLD; and for $\beta \to 0$, it tends to ISD. The choice of the β parameter depends on the statistical distribution of the data, and requires prior knowledge. The utility of the β-divergence cost function is that it enables the selection of many different reconstruction penalty schemes through the selection of a single parameter. In effect, providing a wide selection of NMF algorithms, each producing different results.

NMF is treated as an optimisation problem that minimises the selected cost function, and enforces a non-negativity constraint on the factors:

$$\min_{\mathbf{W}, \mathbf{H}} D_{\mathrm{BD}}(\mathbf{V}\|\mathbf{W}, \mathbf{H}, \beta) \qquad \mathbf{W}, \mathbf{H} \geq 0,$$

resulting in a parts-based decomposition, where the basis in $\mathbf{W}$ resemble parts of the input data, which can only be summed together to approximate $\mathbf{V}$. Eq. 1 is convex in $\mathbf{W}$ and $\mathbf{H}$ individually, but not together. Therefore NMF algorithms usually alternate updates of $\mathbf{W}$ and $\mathbf{H}$. The cost function is minimised using a diagonally rescaled gradient descent algorithm [4], which enforces the non-negativity constraint and leads to the following multiplicative updates for NMF:

$$w_{ij} \leftarrow w_{ij} \frac{\sum_{k=1}^{T} (v_{ik}/[\mathbf{WH}]_{ik}^{2-\beta}) h_{jk}}{\sum_{k=1}^{T} [\mathbf{WH}]_{ik}^{\beta-1} h_{jk}}, \tag{2a}$$

$$h_{jk} \leftarrow h_{jk} \frac{\sum_{i=1}^{M} w_{ij} (v_{ik}/[\mathbf{WH}]_{ik}^{2-\beta})}{\sum_{i=1}^{M} w_{ij} [\mathbf{WH}]_{ik}^{\beta-1}}. \tag{2b}$$

As the NMF algorithm iterates, its non-negative factors converge to a local optimum of Eq. 1.

The parameter R, which specifies the number of columns in $\mathbf{W}$ and rows in $\mathbf{H}$, determines the rank of the approximation. Typically, $R < M$ therefore $\mathbf{W}$ is over-determined and NMF reveals low-rank features of the data. The selection of an appropriate value for R usually requires prior knowledge, and is important in obtaining a satisfactory factorisation.

III Hexaphonic Guitar Transcription Using Non-Negative Constraints

Non-negative matrix factorisation has been previously employed for music transcription with some success [2, 10]. Since NMF is a parts based method it is suitable for polyphonic music transcription, where coinciding notes can be identified and transcribed. Furthermore, NMF makes no prior assumptions about the data, and can be applied blindly to musical recordings from any pitched instrument, where the factorisation yields a basis that resembles notes in the columns of $\mathbf{W}$ and their activations in $\mathbf{H}$.

By virtue of our sophisticated instrument and recording setup, we convert a polyphonic transcription problem into a monophonic one and employ NMF for this simpler problem. Furthermore, transcription of standard single channel guitar recordings is complicated by the fact that there is a repetition of notes throughout the fretboard (which is apparent when tuning a guitar), thus introducing an ambiguity as to what string the note was actually played on (a problem for tablature generation), our setup solves this problem.

Since we are concerned exclusively with hexaphonic guitar transcription, we learn a basis, $\mathbf{W}$, (see Fig. 3) off-line by applying NMF to training data and repeating the process until a satisfactory decomposition that appropriately represents each note is achieved. In contrast, learning $\mathbf{W}$ & $\mathbf{H}$ together, as outlined above, may lead to the discovery of spurious basis elements such as onset components, giving poor results.

a) Transcription Procedure

We perform transcription as six disjoint NMF problems—one for each string—where the inputs to each are constructed using the magnitude spectrogram of the recordings to be transcribed, which are denoted $\mathbf{V}_1, \ldots, \mathbf{V}_6$. Prior to transcription, we apply NMF to the training data and learn a fixed basis, $\mathbf{W}_1, \ldots, \mathbf{W}_6$, for each string, where each column contains a magnitude spectrum that corresponds to the notes particular to that string. Subsequent to basis learning, the columns are arranged in order of ascending pitch. Transcription is achieved by fitting $\mathbf{V}_1, \ldots, \mathbf{V}_6$ to $\mathbf{W}_1, \ldots, \mathbf{W}_6$ using Eq. 2b, resulting in activation matrices $\mathbf{H}_1, \ldots, \mathbf{H}_6$, which indicate the position in time each note is played.

Typically, the output from a transcription algorithm is a *piano roll*[1] transcription, where notes, in ascending order of pitch, are represented on a vertical axis with their activation in time along the horizontal. The form of $\mathbf{H}$ follows this pattern, although the activations are weighted by the volume of the notes being played.

To determine binary (on/off) note durations for the piano roll transcription we apply a maxcol operation to $\mathbf{H}$, which sets all entries to zero and replaces the maximum activation in each column with a one in a winner-takes-all scheme (since at most one note can be played per string), provided that the maximum value is greater than ϵ. To identify note onsets, we mask $\mathbf{H}$ using the output from the maxcol operation and calculate the derivative of the resultant signal, which has peaks that corresponds to onsets in the signal. The peaks are identified by a local maxima *peak-picking* algorithm. After the piano roll is constructed, for validation purposes, we create a MIDI file from the piano roll representation and listen to the results[2].

More formally, we use the following procedure for automatic transcription of hexaphonic guitar recordings:

1. Construct a non-negative representation of the training and test recordings by using the magnitude spectrogram for each string in the recording; we denote as $\mathbf{V}_1, \ldots, \mathbf{V}_6$.

2. Learn a musical note basis, $\mathbf{W}$, for each string by specifying six disjoint NMF problems, Eq. 2, where the training data, $\mathbf{V}_1, \ldots, \mathbf{V}_6$, for each string is factorised separately, and R is dependent on the number of frets. The resultant basis matrices, $\mathbf{W}_1, \ldots, \mathbf{W}_6$, contain in their columns magnitude spectra that resemble the notes of each string, which are then rearranged in order of ascending pitch (see Fig. 3). Rescale each column to the unit L_2-norm, $\mathbf{w}_j = \frac{\mathbf{w}_j}{\|\mathbf{w}_j\|}, j = 1, \ldots, R$.

3. Again specify six disjoint problems and fit $\mathbf{V}_1, \ldots, \mathbf{V}_6$ to $\mathbf{W}_1, \ldots, \mathbf{W}_6$ using the $\mathbf{H}$ update rule, Eq. 2b. Randomly initialise $\mathbf{H}$ and specify β, repeat for the desired number of iterations.

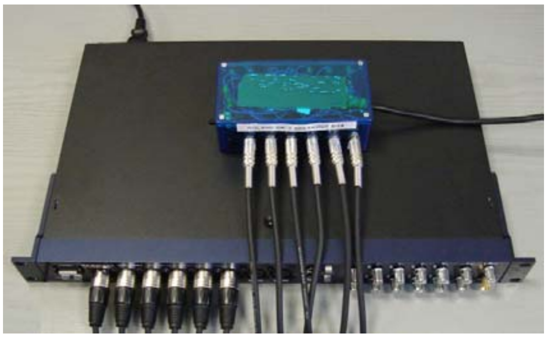
4. Subsequent to fitting, specify ϵ and construct a piano roll representation from $\mathbf{H}_1, \ldots, \mathbf{H}_6$ using the method described.

5. Subjectively evaluate automatic hexaphonic guitar transcription by converting the resultant piano roll representation to a MIDI file and listen to the results.

[1]Named after the pre-programmed perforated paper rolls that stored the music to be played on a late 19$^{\text{th}}$ century mechanical self-playing piano.

[2]We create the necessary functions based on the Matlab toolbox provided at `http://www.kenschutte.com/midi/`

Breakout Box and Recording Equipment



Hexaphonic Guitar

Fig. 1: Bespoke Roland GK-3 divided pickup breakout box connected to a Tascam US-1641 USB audio interface (Left). Roland GK-3 divided pickup installed on a Fender Deluxe Player Stratocaster (right).

IV Hexaphonic Guitar Modification and Recording Procedure

When building a hexaphonic guitar, it may be tempting to modify a standard electric guitar and disassemble its pickup and tap each magnetic polepiece individually, connecting each signal to a separate $\frac{1}{4}$ inch phono jack socket (our initial approach). However, luckily enough, there is a much easier approach using an off-the-shelf Roland GK-3 divided pickup (which is part of the Roland GK MIDI guitar system) and a bespoke breakout box, which serves to connect each of the six string signals that come from the divided pickup unit to a phono jack socket. The breakout (or fanout) box consists of a very simple circuit and is easy to build. Instructions to build the hexaphonic guitar breakout box are available online [11]. Furthermore, the breakout box and hexaphonic guitar has been employed for spatial music performances [12].

We build a hexaphonic guitar by installing the Roland GK-3 divided pickup to a Fender Stratocaster electric guitar, which is then connected to the breakout box. When recording the hexaphonic guitar we connect the six outputs from the breakout box to a Tascam US-1641 USB audio interface (see Fig. 1), which is connected to a laptop computer, and record using Cubase LE music production/recording software. The Tascam unit is designed for multiple microphone recording and has 8 mic/line XLR sockets. Since our breakout box utilises phono sockets, we use $\frac{1}{4}$ inch phono jack to XLR plug cables. The recordings are sampled at 44.1 kHz and stored as 16 bit WAV files. For all recordings the guitar is tuned to standard tuning and the breakout box string outputs are denoted 1–6, which correspond to E(high) B G D A E(low).

a) *Training Data Recording*

For our NMF transcription algorithm we exploit prior knowledge—as we are only concerned with hexaphonic guitar recordings—and learn a basis (or feature set) specific to each string. Since NMF discovers distinct events, delimited by other distinct events, the sequence of the notes in the training data must also appear as distinct events. With this in mind, we use the following sequence to generate our test data: For each string we play the open string and progress up the fretboard playing the string with a plectrum (21 frets on our Stratocaster), pressing on each fret (not slide), after the last note we play the open string again then progress down the fretboard (see Fig. 2). In this way each note is played at least twice and is delimited by different notes each time. The duration for the recordings of each string is 40s on average. As we are only interested in recording one string at this stage, the recordings from the other 5 stings may be discarded.

b) *Test Data Recording*

We make various musical recordings including recordings of chord riffs, solos, leads etc. and use the same setup as for the training data recording. For each string recording we observed almost no leakage signal from the surrounding strings. However, as would be expected, harmonics can be heard on the unplayed strings. Although, the strings resonate at such a low intensity that they can only be heard at high volume and it is not a problem for transcription.

V Automatic Hexaphonic Guitar Transcription Example

We create a non-negative representation for all test and training recordings by constructing a magni-

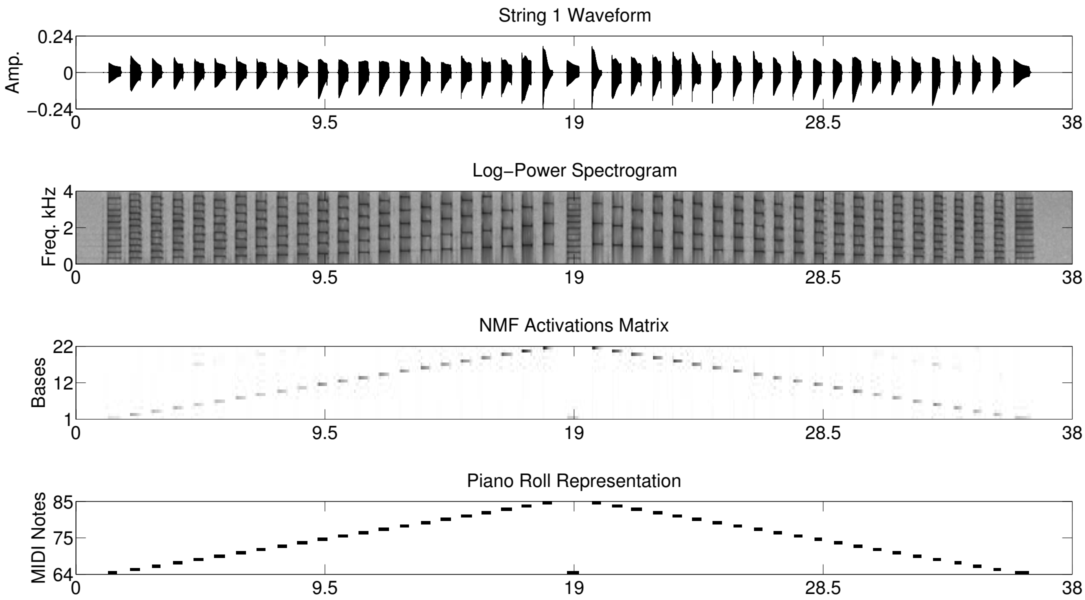


Fig. 2: Illustration of automatic transcription for string 1 using a pre-learned NMF basis. The waveform follows the sequence used for the recording of training data, and the log-power spectrogram indicates the frequency of the notes played. The spectrogram is fitted to the basis using NMF resulting in an activation matrix, which is used to generate a piano roll representation. The piano roll representation indicates that an appropriate basis is learned to transcribe the recording of the string.

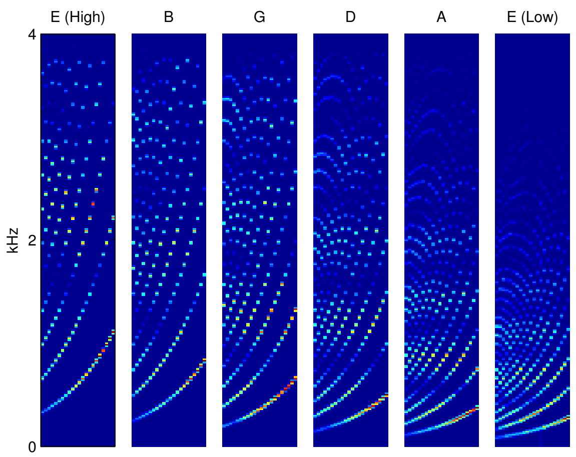


Fig. 3: Magnitude spectrogram representation of the guitar note basis for each string arranged in ascending order according to pitch. The patterns revealed by the partials of each note are indicative of the western scale music *i.e.* logarithmic scale of base $\sqrt[12]{2}$. Strings 1 to 6 appear left to right.

tude spectrogram for each, where we specify a FFT frame size of 512 and a frame overlap of 256. For all experiments we set the NMF parameter $\beta = 1$, *i.e.* KLD, and run NMF for 300 iterations. Furthermore, we downsample recordings to 8 kHz.

a) Hexaphonic Guitar Note Basis

We pre-learn a guitar note basis for each string using the procedure outlined in Section III. However, in order to ameliorate octave-related ambiguities in the basis learning procedure, we divide basis learning into two disjoint NMF problems and learn a separate basis for both octaves on the fretboard, which are subsequently adjoined to form a basis for the string, $\mathbf{W}_1, \ldots, \mathbf{W}_6$ (see Fig. 3). Since NMF may recover non-note components, such as onsets, we set R to a value greater than the number of notes in the recording; non-note components are easily identifiable and are discarded. Subsequent to basis learning, using test data, we observe the transcription performance of each basis as demonstrated in Fig. 2.

b) Hexaphonic Guitar Transcription Example

We present a transcription example of a hexaphonic guitar recording. The recording is a simple musical composition based on the C/G chord and has 3 parts: (1) each string of the chord is played twice in decending order from strings 6 to 1 (2) string 1 is played 3 times at the first fret (the fret closest to the nut) then played open (3) the chord is played once and held for a few seconds.

Transcription is performed using the procedure outlined in Section III, we set $\epsilon = 1$, and results are presented in Fig. 4. The NMF activations matrices, $\mathbf{H}_1, \ldots, \mathbf{H}_6$, indicate the notes played in each recording. However, since there are two octaves per sting there are noticeable weak activations for basis elements that are an octave away from the note played; such octave-related note ambiguities are resolved using the maxcol operator. The resultant piano roll representation successfully reveals the composition of the musical recording, where parts (1) & (3) are perfectly transcribed, and part (2) exhibits only one spurious note. Furthermore, listening to the MIDI file version of the transcription, verifies the quality of the transcription.

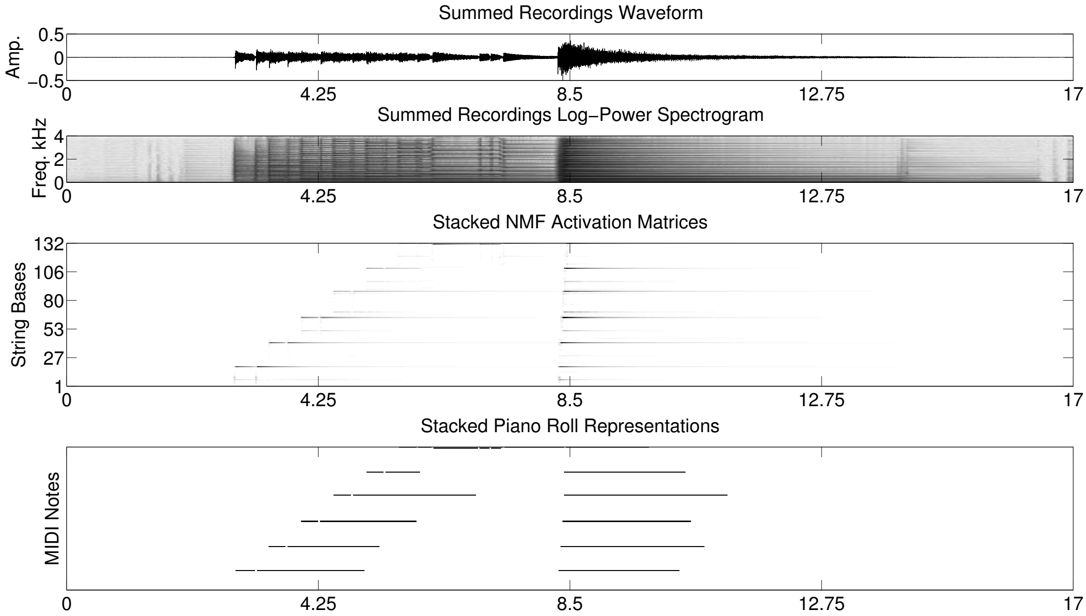


Fig. 4: Automatic transcription of a hexaphonic recording, where the strings of a C/G chord are played individually followed by the full chord. For illustration the 6 recordings are summed and the resultant waveform and log-power spectrogram are presented. The magnitude spectrogram of each string's recording is fitted to a guitar note basis using NMF, resulting in 6 activation matrices, which are stacked and presented. From which, a piano roll representation is constructed (notes on vertical axis are non-sequential as notes overlap between strings), where audible reconstructions verify the transcription.

VI Discussion

We do not exploit note intensity in this work. However, it is possible to combine the note volume information contained in the NMF activation matrices with the piano roll representation to improve the fidelity of MIDI generated music.

Although we only consider electric guitar transcription, it is possible that the proposed approach to transcription may also be used for other stringed instruments that use a pickup for signal capture and subsequent amplification, *e.g.* electric violin.

For hexaphonic guitar transcription, since we analyse each string separately, it is possible to create an ASCII guitar tablature representation of the music, which indicates finger positions rather than musical notes. Although we do not fully explore this extension here, this is our main motivation in this direction.

Since note ambiguities are removed using a hexaphonic recording setup, it is possible to use any electric guitar as a computer interface for music tuition software or video games such as *Guitar Hero*®.

Acknowledgements

We would like to thank Cormac Phelan for his assistance in building the hexaphonic guitar break-out box. This material is based upon works supported by the Science Foundation Ireland under Grant No. 05/YI2/I677.

References

[1] J.P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M.B. Sandler. A tutorial on onset detection in music signals. *Speech and Audio Processing, IEEE Transactions on*, 13(5):1035–1047, Sept. 2005.

[2] Paris Smaragdis and Judith C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 117–180, New Paltz, NY, October 19–22, 2003.

[3] S. A. Abdallah and M. D. Plumbley. Polyphonic transcription by non-negative sparse coding of power spectra. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 318–25, 2004.

[4] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Adv. in Neu. Info. Proc. Sys. 13*, pages 556–62. MIT Press, 2001.

[5] M. N. Schmidt and H. Laurberg. Non-negative matrix factorization with gaussian process priors. *Computational Intelligence and Neuroscience*, 2008.

[6] Paul D. O'Grady and Scott T. Rickard. Automatic ASCII art conversion of binary images using non-negative constraints. In *Proceedings of the Irish Signal and Systems Conference*, pages 186–191, 2008.

[7] D. FitzGerald, M. Cranitch, and E. Coyle. Sound source separation using shifted non-negative tensor factorisation. In *Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006.

[8] Paul D. O'Grady and Barak A. Pearlmutter. Discovering speech phones using convolutive non-negative matrix factorisation with a sparseness constraint. *Neurocomputing*, 72(1–3):88–101, 2008.

[9] Andrzej Cichocki, Rafal Zdunek, and Shun-ichi Amari. Csiszár's divergences for non-negative matrix factorization: Family of new algorithms. In Justinian P. Rosca, Deniz Erdogmus, José Carlos Príncipe, and Simon Haykin, editors, *Independent Component Analysis and Blind Signal Separation, 6th International Conference, ICA 2006, Charleston, SC, USA, March 5-8, 2006, Proceedings*, volume 3889 of *Lecture Notes in Computer Science*, pages 32–9. Springer, 2006.

[10] N. Bertin, R. Badeau, and G. Richard. Blind signal decompositions for automatic transcription of polyphonic music: NMF and K-SVD on the benchmark. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007*, volume 1, pages I–65–I–68, April 2007.

[11] unfretted.net. Breakout box for the Roland GK system. [Online; accessed 13-march-2009].

[12] Enda Bates, Dermot Furlong, and Donnacha Dennehy. Adapting polyphonic pickup technology for spatial music performance. In *Proceedings of the 2008 International Computer Music Conference*, S.A.R.C. Belfast, August 2008.